# Hidden Markov Independent Components Analysis

William D. Penny, Richard M. Everson and Stephen J. Roberts

**Summary.** We propose a generative model for the analysis of nonstationary multivariate time series. The model uses a hidden Markov process to switch between independent components models where the components themselves are modelled as generalised autoregressive processes. The model is demonstrated on synthetic problems and EEG data.

## 1 Introduction

Much recent research in unsupervised learning [17] [20] builds on the idea of using generative models [6] for modelling the probability distribution over a set of observations. These approaches suggest that powerful new data analysis tools may be derived by combining existing models using a probabilistic 'generative' framework. In this paper, we follow this approach and combine hidden Markov models (HMMs), Independent Component Analysis (ICA) and generalised autoregressive models (GAR) into a single generative model for the analysis of nonstationary multivariate time series.

Our motivation for this work derives from our desire to analyse biomedical signals which are known to be highly non-stationary. Moreover, in signals such as the electroencephalogram (EEG), for example, we have a number of sensors (electrodes) which detect signals emanating from a number of cortical sources via an unknown mixing process. This naturally fits an ICA approach which is further enhanced by noting that the sources themselves are characterised by their dynamic content.

The overall generative model is shown in Figure 1. We now look at each component of the model in turn. Section 2 describes the HMM, section 3 describes the ICA model and 3.1 and 3.2 describe the two different source models. Section 4 shows how HMMs and ICA can be integrated, section 5 discusses a number of practical issues and section 6 presents some empirical results.

## 2 Hidden Markov Models

The HMM model has $K$ discrete hidden states, an initial state probability vector $\boldsymbol{\pi} = \pi_1, \pi_2, ..., \pi_K$, a state transition matrix $\boldsymbol{A}$ with entries $a_{ij}$ and

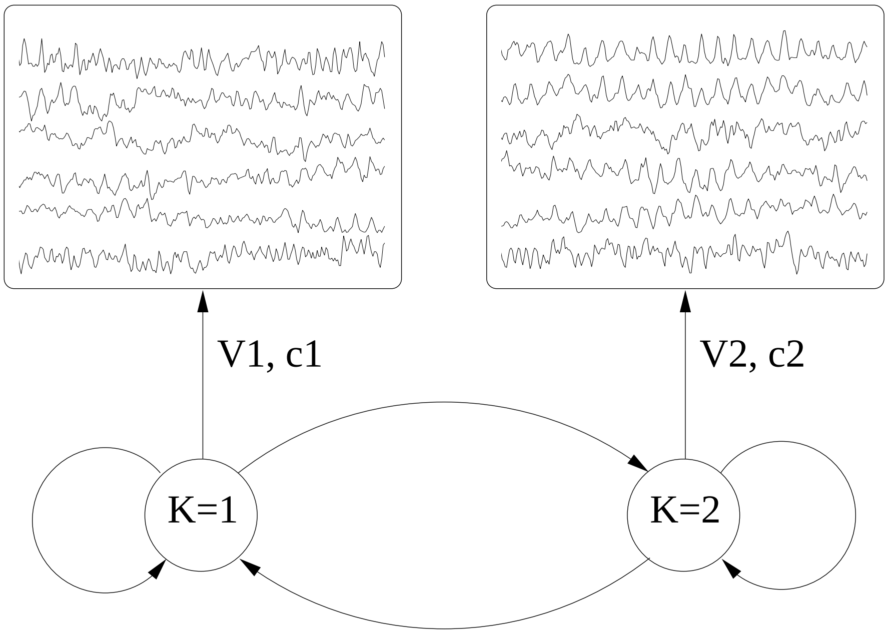


**Fig. 1.** Generative model. At time $t$ the state of the HMM is $q_t = k$. Associated with each state is a mixing matrix $\boldsymbol{V}_k$ and a set of GAR coefficients, $\boldsymbol{c}_k^i$, one for each source $i$. The observations are generated as $\boldsymbol{x}_t = \boldsymbol{V}_k \boldsymbol{a}_t$ and the $i$th source is generated as $a_t^i = \boldsymbol{c}_k^i \tilde{\boldsymbol{a}}_t + e_t$ where $\tilde{\boldsymbol{a}}_t$ is a vector of previous source values and $e_t$ is non-Gaussian noise

an observation density, $b_k(\boldsymbol{x}_t)$, for data point $\boldsymbol{x}_t$ and hidden state $k$. The parameters of the observation model (which in this paper, we consider to be an ICA model) are concatenated into the parameter set $\boldsymbol{B}$. The parameters of the HMM, as a whole, are concatenated into the parameter set $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$. The probability of being in state $k$ at time $t$ given a particular observation sequence $\boldsymbol{X}_1^T$ is $\gamma_k[t]$. Given $\boldsymbol{\theta}$, $\gamma_k[t]$ can be calculated from the forward-backward algorithm [16].

The HMM can be trained using an Expectation-Maximisation (EM) algorithm [5] as follows. In an HMM parameterised by some vector $\hat{\boldsymbol{\theta}}$ the observed variable $\boldsymbol{x}_t$ depends *only* on the current state $q_t$ (and not previous states) and the current state depends *only* on the previous state (and not states before that). This can be seen from the generative model in Figure 1. This allows the joint likelihood of an observation sequence $\boldsymbol{x}_1^T = [\boldsymbol{x}_1, \boldsymbol{x}_t, ..., \boldsymbol{x}_T]$ and hidden variable (state) sequence $\boldsymbol{q}_1^T = [q_1, q_2, ..., q_T]$ to be written as

$$p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q}_1^T, \boldsymbol{x}_1^T) = p_{\hat{\boldsymbol{\theta}}}(q_1) \prod_{t=2}^{T} p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) \prod_{t=1}^{T} p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t \mid q_t) \tag{1}$$

The joint log likelihood may be written

$$\log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q}_1^T, \boldsymbol{x}_1^T) = \log p_{\hat{\boldsymbol{\theta}}}(q_1) + \sum_{t=2}^{T} \log p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) + \sum_{t=1}^{T} \log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t \mid q_t) \tag{2}$$

The EM algorithm requires us to maximise an *auxiliary function*, $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$, the expectation of the joint log likelihood where the expectation is taken relative to the old distribution of hidden (state) variables, $p_{\boldsymbol{\theta}}(\boldsymbol{q}_1^T | \boldsymbol{x}_1^T)$ [1]. Hence,

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) + Q(\boldsymbol{A}, \hat{\boldsymbol{A}}) + Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) \tag{3}$$

where

$$Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) = \sum_{\boldsymbol{q}_1^T} p_{\boldsymbol{\theta}}(\boldsymbol{q}_1^T | \boldsymbol{x}_1^T) \log p_{\hat{\boldsymbol{\theta}}}(q_1) \tag{4}$$

$$Q(\boldsymbol{A}, \hat{\boldsymbol{A}}) = \sum_{\boldsymbol{q}_1^T} p_{\boldsymbol{\theta}}(\boldsymbol{q}_1^T | \boldsymbol{x}_1^T) \sum_{t=2}^{T} \log p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) \tag{5}$$

$$Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) = \sum_{\boldsymbol{q}_1^T} p_{\boldsymbol{\theta}}(\boldsymbol{q}_1^T | \boldsymbol{x}_1^T) \sum_{t=1}^{T} \log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t \mid q_t) \tag{6}$$

and $\sum_{\boldsymbol{q}_1^T}$ denotes a sum over all possible hidden state sequences. The three terms can be maximised separately giving rise to parameter update equations for each different part of the model. Maximising $Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}})$ and $Q(\boldsymbol{A}, \hat{\boldsymbol{A}})$ leads to the usual update equations for the initial state probabilities and the state transition matrix [16]. The term for the observation model parameters can be re-arranged as

$$Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) = \sum_{k} \sum_{t} \gamma_k[t] \log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t \mid q_t) \tag{7}$$

In principle, the observation model can be any probabilistic model. In speech recognition, for example, a Gaussian observation model or Gaussian Mixture model is often used and in previous work we have investigated the use of autoregressive models [14]. In this paper, we consider ICA models.

After the HMM has been trained, it can be applied to novel data sets and the most likely sequence of hidden states calculated using the Viterbi algorithm [16].

[1] Maximising the auxiliary function can be shown, via Baye's rule and Jensen's inequality, to maximise the likelihood of the data, $p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t)$ [5].

## 3 Independent Component Analysis

In Independent Component Analysis (ICA) the observed variable is modelled as

$$\boldsymbol{x}_t = \boldsymbol{V}\boldsymbol{a}_t \tag{8}$$

where the underlying factors or sources, $\boldsymbol{a}_t$, are statistically independent and $\boldsymbol{V}$ is known as the *mixing matrix*. There are $N$ observations and $M$ sources. In this paper we consider square mixing only, i.e. $M = N$. ICA may also be viewed as a linear transformation to a new set of variables

$$\boldsymbol{a}_t = \boldsymbol{W}\boldsymbol{x}_t \tag{9}$$

whose components, $a_i[t]$, are statistically independent. The matrix $\boldsymbol{W}$ is known as the *unmixing matrix* and is the inverse of $\boldsymbol{V}$.

As noted by Mackay [11], it is possible to place ICA in a maximum likelihood framework. The likelihood of data points can be calculated from the likelihood of the source densities by noting that multivariate probability distributions are transformed as follows

$$p(\boldsymbol{x}_t) = \frac{p(\boldsymbol{a}_t)}{|\boldsymbol{J}|} \tag{10}$$

where $||$ denotes the absolute value and $\boldsymbol{J}$ is the Jacobian of the transformation (the Jacobian is the determinant of the matrix of partial derivatives) and $\boldsymbol{a}_t = \boldsymbol{W}\boldsymbol{x}_t$ [12]. Given that, in ICA, the transformation is linear and we are assuming independent sources, we have $|\boldsymbol{J}| = 1/|det(\boldsymbol{W})|$ and $p(\boldsymbol{a}_t) = \prod_i a_i[t]$. This allows us to write the log likelihood as

$$\log p(\boldsymbol{x}_t) = \log |det(\boldsymbol{W})| + \sum_{i=1}^{M} \log p(a_i[t]) \tag{11}$$

where $\boldsymbol{W}$ is the unmixing matrix and $p(a_i[t])$ is the probability density of the $i$th source component and $a_i[t] = \sum_j \boldsymbol{W}_{ij} x_j[t]$. We now describe two different source density models.

### 3.1 Generalised Exponential Sources

In Bell and Sejnowski's original ICA formulation [2] the source densities were (implicitly) assumed to be inverse-cosh densities (see [11] for a discussion). More recent work (see, for example, [7]) however, shows that the inverse-cosh form cannot adequately model sub-Gaussian densities[2]. A more general parametric form which can model super-Gaussian, Gaussian *and* sub-Gaussian

[2] A sub-Gaussian density has negative kurtosis, a Gaussian density zero kurtosis, and a super-Gaussian density positive kurtosis. Sub/super densities have lighter/heavier tails than Gaussians.

forms is the 'Exponential Power Distribution' or 'Generalised Exponential' (GE) density

$$G(a; R, \beta) = \frac{R\beta^{1/R}}{2\Gamma(1/R)} \exp(-\beta|a|^R) \tag{12}$$

where $\Gamma()$ is the gamma function [15]. This density has zero mean, a kurtosis determined by the parameter $R$ and a variance which is then determined by $1/\beta$; see Appendix A for further details. This source model has been used in an ICA model developed by Everson and Roberts [7]. The unmixing matrix, $\boldsymbol{W}$, can be estimated using the usual covariant-based algorithms, or second-order methods such as BFGS, and the source density parameters, $\{R, \beta\}$, can be estimated using an embedded line search [7].

With an explicit model for the source density, such as a generalised exponential, it is convenient for interpretation purposes for the unmixing matrix to be row-normalised. This allows the *directions* of the unmixing to be solely contained in $\boldsymbol{W}$ and the *magnitudes* of the sources to be solely contained in the source density parameters, eg. $R$ and $\beta$. For all of the simulations in this paper we use row normalisation.

### 3.2 Generalised Autoregressive Sources

Pearlmutter and Parra [13] have proposed a 'contextual-ICA' algorithm where the sources are conditioned on previous source values. The observations are then generated from an instantaneous mixing of the sources. Their work focuses on using generalised autoregressive (GAR) models for modelling each source. The term 'generalised' is used because the AR models incorporate additive noise which is non-Gaussian. Specifically, Pearlmutter and Parra use an inverse-cosh noise distribution.

Pearlmutter and Parra have shown that contextual-ICA can separate sources which cannot be separated by standard (non-contextual) ICA algorithms. This is because the standard methods utilise only information from the cumulative histograms; temporal information is discarded.

These methods are particularly suited to EEG, for example, as sources are identified both by their spatial location *and* by their temporal content eg. frequency characteristics.

In this paper we will use contextual information by modelling the sources as Generalised Autoregressive GAR models with $p$ filter taps and additive noise drawn from a generalised exponential distribution; for $p = 0$ these models therefore reduce to the GE sources described in the previous section. The density model is

$$p(a_i[t]) = G(e_i[t], R_i, \beta_i) \tag{13}$$

where $e_i[t] = a_i[t] - \hat{a}_i[t]$ is the GAR prediction error and $\hat{a}_i[t]$ is the GAR prediction

$$\hat{a}_i[t] = -\sum_{d=1}^{p} c_i[d] a_i[t-d] \tag{14}$$

where $c_i[d]$ are the GAR coefficients for the $i$th source which can collectively be written as a vector $\boldsymbol{c}_i$.

## 4 Hidden Markov ICA

A Hidden Markov ICA model has an unmixing matrix, $\boldsymbol{W}_k$, and source density parameter vectors, $\boldsymbol{R}_k, \boldsymbol{\beta}_k$ for each state $k$. These parameters can be optimised by minimising a cost function defined by substituting the ICA log-likelihood (equation 11) into the HMM auxiliary function (equation 7)

$$Q = \sum_k Q_k \tag{15}$$

where

$$Q_k = \log |det(\boldsymbol{W}_k)| + \frac{1}{\gamma_k} \sum_t \gamma_k[t] \sum_i \log p(a_i[t]) \tag{16}$$

where we have divided through by

$$\gamma_k = \sum_t \gamma_k[t] \tag{17}$$

Gradient-based learning rules for updating the parameters of the HMICA model can be derived by evaluating the appropriate derivatives of $Q_k$ eg. $\frac{\partial Q_k}{\partial \boldsymbol{W}_k}$, $\frac{\partial Q_k}{\partial R_{k,i}}$, $\frac{\partial Q_k}{\partial \beta_{k,i}}$ and $\frac{\partial Q_k}{\partial \boldsymbol{c}_{k,i}}$. The remainder of this section shows how these derivatives can be computed. The unmixing matrix can be learnt by following the gradient

$$\frac{\partial Q_k}{\partial \boldsymbol{W}_k} = \boldsymbol{W}_k^{-T} + \frac{1}{\gamma_k} \sum_t \gamma_k[t] \boldsymbol{z}_t \boldsymbol{x}_t^T \tag{18}$$

where $z_i(t) = \partial \log p(a_i[t]) / \partial a_i[t]$. It is, however, more efficient to use a 'covariant' [11] algorithm which follows the 'natural gradient' [1] which is obtained by post-multiplying the gradient by $\boldsymbol{W}_k^T \boldsymbol{W}_k$. This gives

$$\left[\frac{\partial Q_k}{\partial \boldsymbol{W}_k}\right]_{NAT} = \boldsymbol{W}_k + \frac{1}{\gamma_k} \sum_t \gamma_k[t] \boldsymbol{z}_t \boldsymbol{y}_t^T \tag{19}$$

where

$$\boldsymbol{y}_t = \boldsymbol{W}_k^T \boldsymbol{a}_t \tag{20}$$

The above gradients are identical to those used in standard ICA except that, in the ICA update for state $k$, each data point is weighted by an amount proportional to the 'responsibility' state $k$ has for that data point, $\gamma_k[t]/\gamma_k$.

In what follows, there is an unmixing matrix for each state and there are source model parameters for each source and for each state. But to keep the notation simple, we drop any references to the state $k$ (eg. we write $\boldsymbol{W}_{ij}$ not $\boldsymbol{W}_{k,ij}$, $R_i$ not $R_{k,i}$ and $\beta_i$ not $\beta_{k,i}$). We now derive HMICA learning rules for the two cases of (i) GE sources and (ii) GAR sources.

### 4.1 Generalised Exponential Sources

The natural gradient is given by substituting the derivative of the GE density into the general weight update equation derived above. We therefore substitute $z_i(t)$ from equation 35 (in the appendix) into equation 19. This gives

$$\left[\frac{\partial Q_k}{\partial \boldsymbol{W}_{ij}}\right]_{NAT} = \boldsymbol{W}_{ij} - \frac{1}{\gamma_k}\sum_t \gamma_k[t]sign(a_i[t])R_i\beta_i|a_i[t]|^{R_i-1}y_j[t] \tag{21}$$

where

$$y_j[t] = \sum_i \boldsymbol{W}_{ij}a_i[t] \tag{22}$$

To update the parameters of the GE sources the appropriate likelihood function can found by substituting $\log p(a_i[t])$ from equation 34 (in the appendix), for each source in turn, into equation 16. This gives

$$Q_k = \log|det(\boldsymbol{W}_k)| + \frac{1}{\gamma_k}\sum_t \gamma_k[t]\sum_i \log\left[\frac{R_i\beta_i^{1/R_i}}{2\Gamma(1/R_i)}\right] - \beta_i|a_i(t)|^R \tag{23}$$

The source density parameters for each state can now be estimated componentwise by evaluating the derivatives $\frac{\partial Q_k}{\partial \beta_i}$ and $\frac{\partial Q_k}{\partial R_i}$, where $i$ refers to the $i$th source, and setting them to zero. This gives

$$\frac{1}{\beta_i} = \frac{R_i}{\gamma_k}\sum_t^T \gamma_k[t]|a_i[t]|^{R_i} \tag{24}$$

and $R_i$ can be found by solving the one-dimensional problem

$$\frac{\partial Q_k}{\partial R_i} = \frac{1}{R_i} + \frac{1}{R_i^2}\log\beta_i + \frac{1}{R_i^2}\varphi(1/R_i) - \frac{\beta_i}{\gamma_k}\sum_{t=1}^T \gamma_k[t]|a_i[t]|^{R_i}\log|a_i[t]| = 0 \tag{25}$$

where $\varphi(x) = \Gamma'(x)/\Gamma(x)$ is the digamma function [15]. Equations 24 and 25 are generalised versions of the equations in Appendix A of [7] and are applied iteratively until a consistent solution is reached.

### 4.2 Generalised Autoregressive Sources

For GAR sources the analysis is identical to that described for GE sources except that, in equations 21 to 25 (but not equation 22), $a_i[t]$ is replaced by $e_i[t]$ where $e_i[t] = a_i[t] - \hat{a}_i[t]$ and $\hat{a}_i[t]$ is the GAR prediction. The GAR coefficients, $\boldsymbol{c}_i$, can be found separately for each source by minimising

$$E_{k,i} = \frac{1}{\gamma_k} \sum_t^T \gamma_k[t] |a_i[t] - \hat{a}_i[t]|^{R_i} \tag{26}$$

which can be achieved by descending the gradient $\frac{\partial E_{k,i}}{\partial \boldsymbol{c}_i}$. For the $d$th element of $\boldsymbol{c}_i$ this gives

$$\frac{\partial E_k}{\partial c_{d,i}} = \frac{1}{\gamma_k} \sum_t \gamma_k[t] a_i[t-d] R_i sign(e_i[t]) |e_i[t]|^{R_i - 1} \tag{27}$$

We use this gradient in a Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimizer [15].

## 5 Practical Issues

### 5.1 Initialisation

As learning is based on a maximum likelihood framework, and there may be a number of local maxima in the likelihood landscape, our final solution may be sensitive to the initial parameter settings.

Domain knowledge can be used to initialise some of the parameters. For example, it might be known that some states persist for a stereotypical time period. This information can be used to initialise the state transition matrices. It may also be known that the sources we are interested in have a particular spectral content or are in a particular location. This information can be used to initialise the GAR source models or ICA unmixing matrices.

Alternatively, we could use methods employed in previous research [14] where a first pass through the data and a subsequent cluster analysis identifies good initial conditions for the observation models.

We may also have domain knowledge which requires that some parameters be tied or fixed to certain values. For example, it may be known that the sources do not change but the mixing matrices do or vice-versa.

### 5.2 Learning

The pseudo-code in figure 2 shows the three layers of inference in the HMICA model with GAR sources. The three nested loops (HMM optimisation, ICA optimisation and GAR optimisation) require convergence criteria which are

```
Initialise HMM transition matrix, A, and initial probability vector, π.

Initialise ICA mixing matrices, W_k, and source parameters c_i, R_i and β_i.

While HMM updates not converged,

    Run forward-backward algorithm to calculate γ_k[t].

    for k = 1 to K,

        While ICA updates not converged,

            Get source values, predicted source values and errors:

            a_t = W x_t, â_i[t] = −Σ_{d=1}^{p} c_i[d] a_i[t − d], e_i[t] = a_i[t] − â_i[t]

            Update unmixing matrix:

            ΔW_ij ∝ −(1/γ_k) Σ_t γ_k[t] sign(e_i[t]) R_i β_i |e_i[t]|^(R_i − 1) y_j[t]

            If ICA iteration number is divisible by S,

                Update GAR coefficients:

                Δc_{i,d} ∝ (1/γ_k) Σ_t γ_k[t] a_i[t − d] R_i sign(e_i[t]) |e_i[t]|^(R_i − 1)

                Update GAR noise model:

                1/β_i = (R_i/γ_k) Σ_t^T γ_k[t] |e_i[t]|^(R_i). Also update R_i.

            end

        end

    end

    Update HMM state transition matrix and initial probability vector.

end
```

Equations in the pseudo-code, in LaTeX:

$$\boldsymbol{a}_t = \boldsymbol{W}\boldsymbol{x}_t, \hat{a}_i[t] = -\sum_{d=1}^{p} c_i[d]a_i[t-d], e_i[t] = a_i[t] - \hat{a}_i[t]$$

$$\Delta \boldsymbol{W}_{ij} \propto -\frac{1}{\gamma_k}\sum_t \gamma_k[t] sign(e_i[t]) R_i \beta_i |e_i[t]|^{R_i - 1} y_j[t]$$

$$\Delta c_{i,d} \propto \frac{1}{\gamma_k}\sum_t \gamma_k[t] a_i[t-d] R_i sign(e_i[t]) |e_i[t]|^{R_i-1}$$

$$\frac{1}{\beta_i} = \frac{R_i}{\gamma_k}\sum_t^T \gamma_k[t]|e_i[t]|^{R_i}$$

**Fig. 2.** Pseudo-code for learning in a hidden Markov ICA model with GAR sources

defined as follows. If $LL(new)$, $LL(old)$ and $LL(init)$ are the old, new and initial log-likelihood values then convergence is defined by

$$\frac{LL(new) - LL(init)}{LL(old) - LL(init)} < 1 + \delta \tag{28}$$

For the HMM loop we use $\delta = 10^{-4}$ and for the ICA and GAR loops we use $\delta = 10^{-2}$. A further issue is the choice of $S$, the number of updates of the ICA unmixing matrix before the source model is re-estimated. In this paper we set $S = 10$. Optimisation does not, however, seem to be particularly sensitive to the choice of these parameters.

### 5.3 Model order selection

We have to decide on the number of HMM states, $K$, the number of source variables, $M$, and the number of GAR parameters in each source model, $p$. In general, we rely on domain knowledge, although generic methods do exist, for example, for GAR model order selection [4] and HMM model order selection [18]. For the ICA model, as we consider only the case of square mixing, model order selection is not an issue, although we discuss this further in section 7.

## 6 Results

In section 6.1 we compare the GE and GAR source models on a synthetic data set. In 6.2 and 6.3 we demonstrate that the HMICA model can partition multiple time series on the basis of having different underlying source models or mixing matrices or both. In 6.4 we apply the HMICA model to EEG data.

### 6.1 Multiple sinewave sources

In this section we compare the GE and GAR source models on a synthetic data set. The data consists of two sources, one composed of three sine waves of unit amplitude (ranging from +1 to -1) at frequencies of 5, 15 and 25Hz plus additive Laplacian noise, and the other composed of two sine waves of unit amplitude at frequencies of 10 and 20Hz plus additive Gaussian noise. The two sources were then mixed using

$$\boldsymbol{V} = \begin{pmatrix} 0.8321 & 0.5547 \\ 0.4472 & 0.8944 \end{pmatrix} \tag{29}$$

and $N = 200$ samples were generated in this way [3]. The quality of the unmixing was measured by the unmixing error

$$E = \min_{\boldsymbol{P}} \frac{||\boldsymbol{WVP} - diag(\boldsymbol{WVP})||}{||\boldsymbol{WV}||} \tag{30}$$

[3] If the sampling rate is set to be a multiple of any of the frequencies in the signal this can result in pathological sampling; care was therefore taken to avoid this by choosing the sampling rate to be $60 + \pi$Hz, an irrational number.

where $\boldsymbol{W}$ is our estimated unmixing matrix and $\boldsymbol{P}$ is a permutation matrix [7]. The variance of the source noise was then set so as to achieve a certain Signal-to-Noise Ratio (SNR), which was the same for both sources. Table 1 shows the unmixing errors for the two methods over a range of SNRs.

**Table 1.** Comparison of GA and GAR source models. Unmixing errors, $E$, for Generalised Exponential (GE) and Generalised Autoregressive (GAR) source models on multiple sine wave sources. Each entry is the mean value from 20 runs with $\pm\sigma$ indicating the standard deviation. At high source SNR the GAR source model provides better unmixing

| SNR | $E_{GE}$ | $E_{GAR}$ |
|---|---|---|
| 0.01 | 0.09 $\pm$ 0.05 | 0.13 $\pm$ 0.07 |
| 0.10 | 0.08 $\pm$ 0.05 | 0.10 $\pm$ 0.07 |
| 1.00 | 0.13 $\pm$ 0.09 | 0.06 $\pm$ 0.04 |
| 10.0 | 0.19 $\pm$ 0.29 | 0.02 $\pm$ 0.01 |

The results show that, as the source SNR increases, the GAR source model results in better unmixing; at $SNR = 10$ the unmixing is improved by a factor of 10 [4] . The reason for this comparative improvement is two-fold; firstly, the GAR source model itself does a better job and, secondly, and more significantly, the GE model breaks down at high SNR. This break-down is attributed to the fact that, at high SNR, a signal consisting of multiple sinusoids has a multimodal PDF [5]. This multimodality can also be seen empirically in Figure 3. Therefore, the unimodal GE distribution is not capable of modelling high SNR sources. The GAR model, on the other hand, fits the dynamics of the source signals leaving a unimodal residual which is adequately modelled by its GE noise model.

Knuth's empirical work [10] suggests that many EEG sources have multimodal PDFs. For such cases, we would therefore expect GAR source models to provide better unmixing than GE source models. Ultimately, as the GE source model is a special case of GAR with $p = 0$, the choice of which is more appropriate is a model order selection problem.

[4] At low SNR, we effectively have a Laplacian source and a Gaussian source which can be separated by ICA with a GE source model [7]; if *both* sources were Gaussian, however, then this would not be the case.

[5] The PDF of a single sinusoid $R \sin wt$ has modes at $\pm R$ [12], and the addition of further sinusoids of different amplitude and frequency can result in further modes. This PDF is then convolved with the noise PDF to produce a somewhat smeared density where the amount of smearing depends on the noise variance.

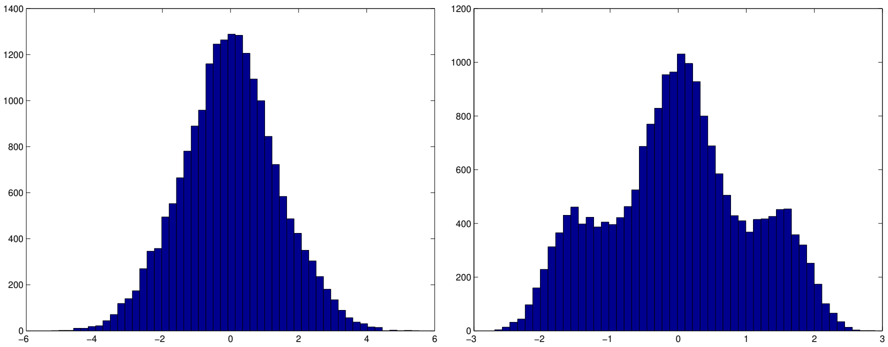


**Fig. 3.** Multimodality of multiple sinewave sources. Empirical PDF (N=20,000) of source 2 at SNR=1 (*left plot*) and SNR=10 (*right plot*). At lower SNRs the noise acts to smear out the peaks resulting in a unimodal PDF

### 6.2 Same sources, different mixing

We generated a two-dimensional time series containing two epochs each with $N = 200$ samples. We first generated two GE sources, the first being Gaussian ($R = 2$) with variance $1/\beta = 2$, and the second being Laplacian ($R = 1$) with width parameter $1/\beta = 1$. These source parameters were the same for both epochs. Observations for the first epoch of the time series were then generated by mixing the sources with the matrix

$$\boldsymbol{V}_1 = \begin{pmatrix} 0.8321 & 0.5547 \\ 0.4472 & 0.8944 \end{pmatrix} \tag{31}$$

In the second epoch, the matrix was changed to

$$\boldsymbol{V}_2 = \begin{pmatrix} 0.8944 & -0.4472 \\ -0.5547 & 0.8231 \end{pmatrix} \tag{32}$$

The resulting data set is plotted in Figure 4. We then trained two ICA models on each separate data partition to serve as a benchmark against which the HMICA solution could be compared[6].

A two-state HMICA model was then trained on the data. The model was initialised by setting $R = 2$ and $\beta = 1$ for both sources and both states. Each unmixing matrix was initially set to the identity matrix with uniform noise in the range $[-0.05, 0.05]$ added on to each matrix element. The diagonal elements of the state transition matrix were set so that the average state

[6] If we were to compare the HMICA solution against the known true models then any difference could be due to (i) incorrect partitioning in the HMICA model or (ii) having a small, finite data sample or (iii) some artifact of the covariant ICA algorithm. By comparing against the covariant ICA solutions we are therefore sure to be focussing only on (i).

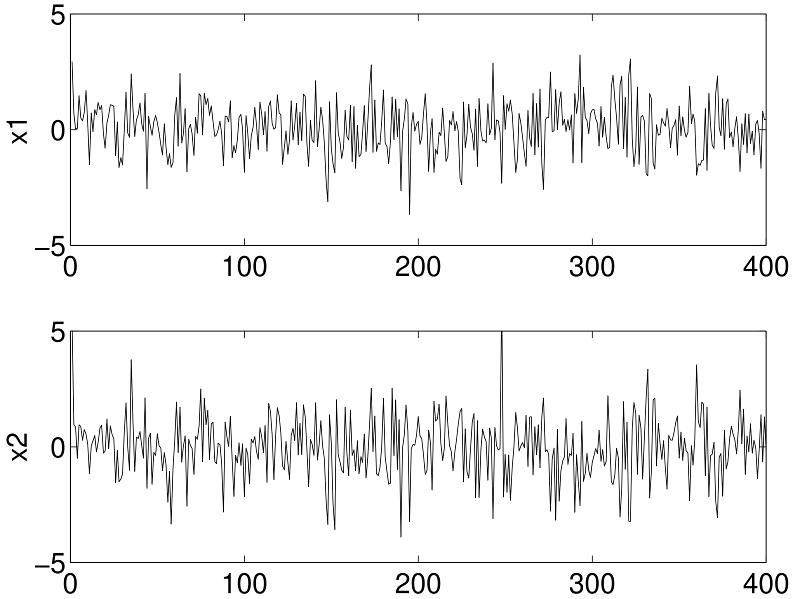


**Fig. 4.** Same sources, different mixing. The mixing matrices underlying the two time series are switched at t=200

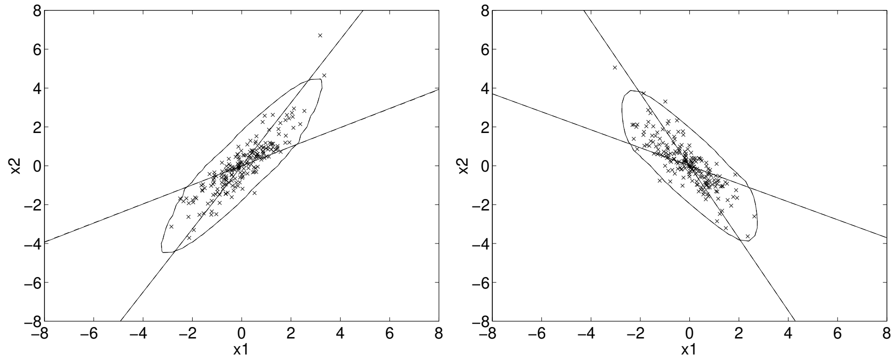


**Fig. 5.** Same sources, different mixing. ICA models for state 1 (*left plot*) and state 2 (*right plot*). The straight lines show the directions of the independent components and the curvy lines show a single probability contour plotted at 0.01 of the probability density at the origin. These probability densities were calculated using equation 10. The solid lines are from individual ICA models trained on the correct partitioning of the data and the dotted lines are estimates from the HMICA model (the dotted lines are barely visible as the estimates overlap almost exactly with the true values)

duration density was $D_s$ samples (see below). Five iterations of the EM algorithm (using equations 21, 24 and 25 to update the ICA unmixing matrices and source density parameters) were then sufficient for the HMICA solution to converge. The estimated ICA models are compared to the benchmark ICA models in Figure 5. The almost complete overlap of true and estimated probability contours shows that $R$ and $\beta$ are estimated very accurately. This accuracy is due to the correct partitioning of the data into the two separate epochs; application of the Viterbi algorithm led to 100% of samples being correctly classified. Experimentation with initialisation of the state transition matrix showed that convergence was typically achieved in five to ten EM iterations with $D_s$ ranging from 10 to 400 samples. Convergence was even possible with $D_s$ as low as three, after ten to twenty EM iterations.

### 6.3 Same mixing, different sources

We generated a two-dimensional time series containing two epochs each with $N = 200$ samples. Data for the first epoch was identical to the first epoch in the last example. Data for the second epoch used the same mixing matrix but the source densities were both chosen to be uniform.

The resulting data set is plotted in Figure 6. We then trained two ICA models on each separate data partition to serve as a benchmark against which the HMICA solution could be compared. A two-state HMICA model was then trained on the data after initialising using the same method as in the previous example. Seven iterations of the EM algorithm were then sufficient for the HMICA solution to converge. The estimated ICA models are compared to the benchmark ICA models in Figure 7. Application of the Viterbi algorithm led to 100% of samples being correctly classified.

### 6.4 EEG data

We now apply the HMICA model to EEG data recorded whilst a subject performed voluntary hand movements in response to cues. Figure 8 shows two channels of EEG and the timing of the movement cue. This subject possessed a particularly strong mu-rhythm (an 8-12Hz rhythm in sensorimotor cortex) which disappeared during hand-movement (after the cue).

We trained two HMICA models on this data set; one with GAR sources, one with GE. Each converged within ten EM iterations. As can be seen from Figure 8 the one with GAR sources provides a reasonable partitioning of the data, whereas use of the GE source model provides a less clear partitioning. This difference can be explained by the high SNR during state 2 (see section 6.1); for the GAR sources in state 2, the SNR's are 3.4 and 5.5.

We also applied the HMICA model to EEG data recorded whilst a subject performed a motor imagery task; the subject was instructed to imagine opening and closing his hand in response to an external cue. Figure 9 shows the EEG data and the resulting partitioning from two HMICA models. This

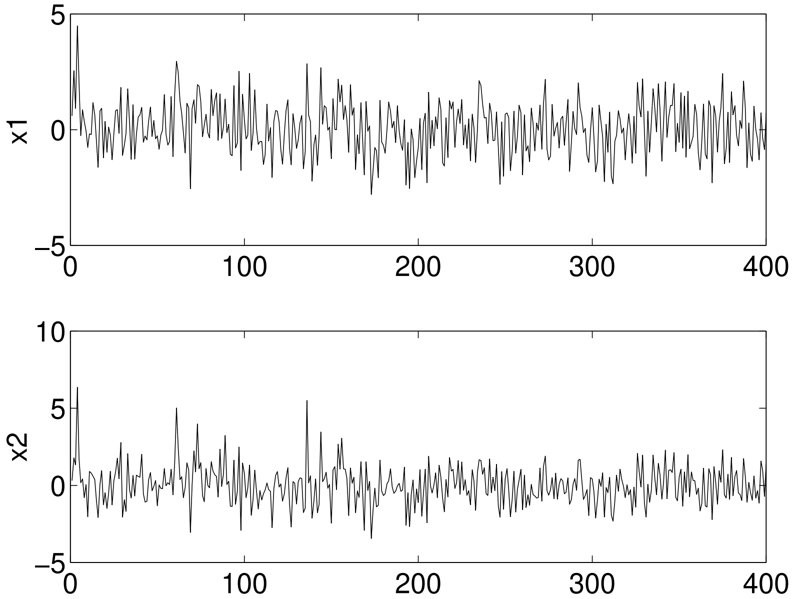


**Fig. 6.** Same mixing, different sources. The source models underlying the two time series are switched at t=200

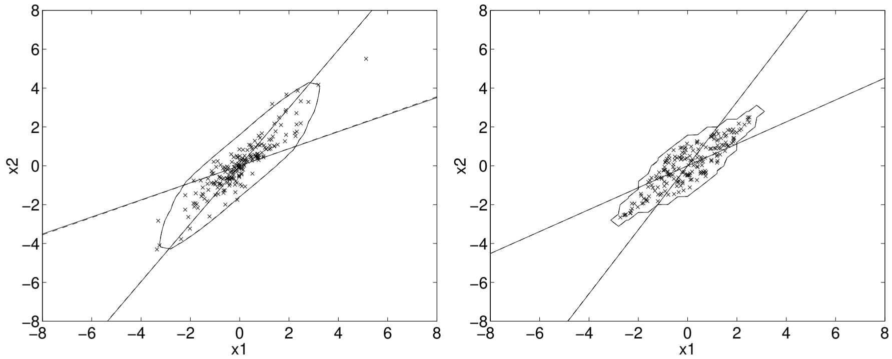


**Fig. 7.** Same mixing, different sources. ICA models for state 1 (*left plot*) and state 2 (*right plot*). The straight lines show the directions of the independent components and the curvy lines show a single probability contour plotted at 0.01 of the probability density at the origin. The solid lines are from individual ICA models trained on the correct partitioning of the data and the dotted lines are estimates from the HMICA model (the dotted lines are barely visible as the estimates overlap almost exactly with the true values)

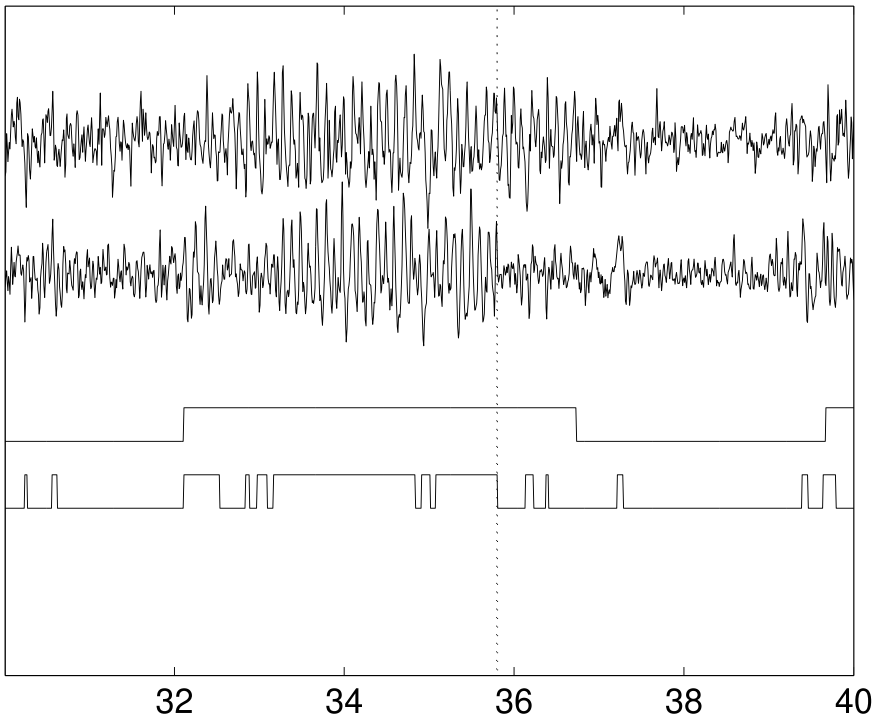


**Fig. 8.** EEG data recorded before and after motor task. The top two traces show EEG recordings from electrodes at positions C3 and C4 (10-20 System). The next two traces show partitionings from an HMICA model; the top one using GAR sources and the bottom one using GE sources. The dashed vertical line indicates a cue telling the subject when to begin hand movement

again demonstrates that the HMICA model can partition time series into meaningful stationary regimes and that GAR source models may often be more appropriate than GE source models.

We note that it is essential to have a nonstationary model in order for these periods of strong (high SNR) rhythmic source activity to be identified. Applying stationary-ICA to the time series without the partitioning provided by the HMM misses this activity completely. GAR source models and HMM switching are therefore complementary.

## 7 Discussion

We have proposed a generative model for the analysis of nonstationary multivariate time series. The model combines three existing statistical models, HMMs, ICA and GARs, into a single overall model where the components

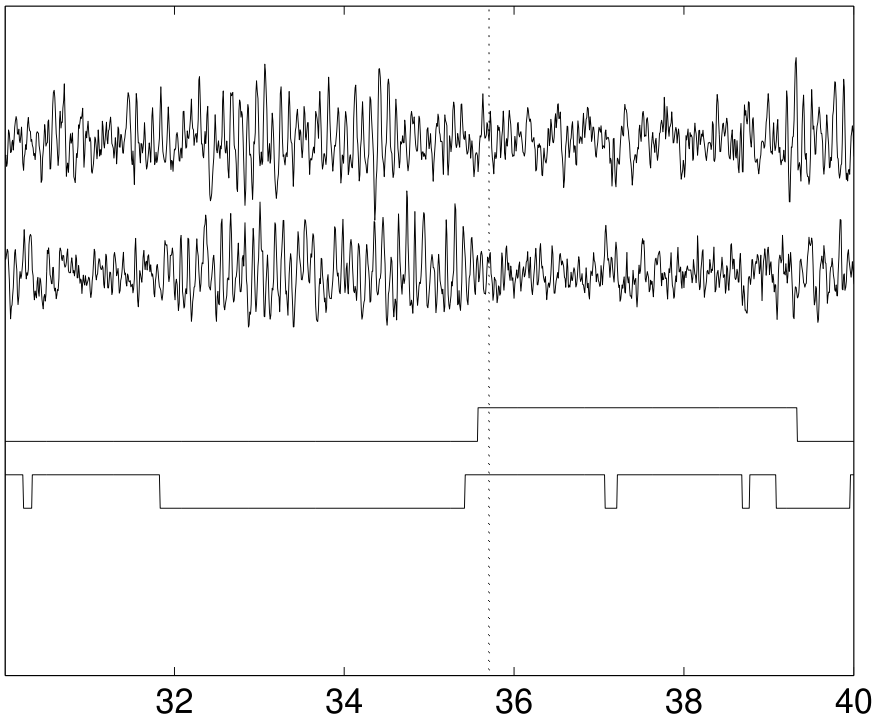


**Fig. 9.** EEG data recorded before and after motor imagery task. The top two traces show EEG recordings from electrodes at positions C3 and C4 (10-20 System). The next two traces show partitionings from an HMICA model; the top one using GAR sources and the bottom one using GE sources. The dashed vertical line indicates the cue indicating when the task should be performed

learn together. The model can (i) learn the dynamics of the underlying sources (ii) learn the parameters of the mixing process and (iii) detect discrete changes in either the source dynamics or the mixing or both. This last property enables the model to partition a multivariate time series into different regimes and so may be used for classification purposes. Moreover, each regime is characterised by a known mixing of stationary sources. The model is therefore readily interpretable.

As compared to previous work in ICA, the main novelty of our approach is in the use of an HMM to switch between ICA models. Like other modellers [13], we have also used GAR source models, although ours are slightly different in the choice of noise model; instead of an inverse-cosh we use the more flexible generalised exponential [7].

Whilst our model goes some way to bridging the gap between statistical models and the known properties of biomedical data, such as EEG, there remain many areas for refinement. The most unrealistic assumption is that of square mixing; for data such as EEG it is likely that the number of sources is less than the number of electrodes. A non-square ICA algorithm is therefore required, such as those developed by Attias[8][9] . These algorithms also have the advantage of being able to cope with observation noise (actually, the observation noise is essential in developing a probabilistic model; if you have fewer sources than observations the 'extra' observations can be explained as noise).

The second main area for improvement is the use of automatic model order selection methods; both for the order of the source model and to choose the optimal number of sources. Again, Attias' algorithms can do this. Finally, following Knuth [10], we could use our knowledge of the physics of the mixing process to choose suitable priors for the mixing matrices.

## 8 Acknowledgements

William Penny is supported by the UK Engineering and Physical Sciences Research Council grant number GR/M 41339. The empirical work in this paper was based around hidden Markov model software routines made available by Zoubin Ghahramani ($http://gatsby.ucl.ac.uk/\tilde{}zoubin$).

## Appendix

The 'exponential power' or 'generalised exponential' probability density is defined as

$$p(a) = G(a; R, \beta) = \frac{R\beta^{1/R}}{2\Gamma(1/R)} \exp(-\beta|a|^R) \tag{33}$$

where $\Gamma()$ is the usual gamma function [15], the mean of the distribution is zero, the width of the distribution is determined by $1/\beta$ and the weight of its tails is set by $R$. This gives rise to a Gaussian distribution for $R = 2$, a Laplacian for $R = 1$ and a uniform distribution in the limit $R \to \infty$. The log-likelihood is given by

$$\log p(a) = \log \left[ \frac{R\beta^{1/R}}{2\Gamma(1/R)} \right] - \beta|a|^R \tag{34}$$

The partial derivative, $z(a) = \frac{\partial log p(a)}{\partial a}$, is given by

$$z(a) = -sign(a) R\beta |a|^{R-1} \tag{35}$$

We also note that the density is equivalently parameterised by a variable $w$ where $w = \beta^{-1/R}$ giving

$$p(a) = \frac{R}{2w\Gamma(1/R)} \exp(-|a/w|^R) \tag{36}$$

and that the density is unimodal and symmetric with variance given by

$$V = w^2 \frac{\Gamma(3/R)}{\Gamma(1/R)} \tag{37}$$

which for $R = 2$ gives $V = 0.5w^2$. The kurtosis is given by [3]

$$K = \frac{\Gamma(5/R)\Gamma(1/R)}{\Gamma(3/R)^2} - 3 \tag{38}$$

Samples may be generated from the density using a rejection method [19].

## References


1. S. Amari, A. Cichocki, and H.H. Yang. A New Learning Algorithm for Blind Signal Separation. In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*. MIT Press, Cambridge, MA, 1996.
2. A. J. Bell and T. J. Sejnowski. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.
3. P. Bratley, B. L. Fox, and E. L. Schrage. *A Guide to Simulation*. Springer, 1983.
4. C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapman and Hall, 1996.
5. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.
6. B. S. Everitt. *An Introduction to Latent Variable Models*. Chapman and Hill, London and New York, 1984.
7. R. Everson and S.J. Roberts. Independent Component Analysis: A Flexible Nonlinearity and Decorrelating Manifold Aproach. *Neural Computation*, 11(8), 1999.
8. H. Attias. Independent Factor Analysis. *Neural Computation*, 11(4):803–851, 1999.
9. H. Attias. Inferring Parameters and Structure of Latent Variable Models by Variational Bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.
10. K. H. Knuth. Difficulties Applying Recent Blind Source Separation Techniques to EEG and MEG. In J.T. Rychert and C.R. Smith, editors, *Maximum Entropy and Bayesian Methods*, pages 209–222. Dordrecht, 1998.
11. D.J.C. Mackay. Maximum Likelihood and Covariant Algorithms for Independent Component Analysis. Technical report, Cavendish Laboratory, University of Cambridge, 1996.
12. A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.
13. B. A. Pearlmutter and L.C. Parra. Maximum Likelihood Blind Source Separation: A Context-Sensitive Generalization of ICA. In *Advances in Neural Information Processing Systems 9*, pages 613–619. MIT Press, 1997.

14. W.D. Penny and S.J. Roberts. Dynamic Models for Nonstationary Signal Segmentation. *Computers and Biomedical Research*, 1999. To appear.
15. W. H. Press, S.A. Teukolsky, W.T. Vetterling, and B.V.P. Flannery. *Numerical Recipes in C, second edition.* Cambridge, 1992.
16. L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
17. S. Roweis and Z. Ghahramani. A Unifying Review of Linear Gaussian Models. *Neural Computation*, 11(2):305–346, 1999.
18. T. Ryden. Estimating the Order of Hidden Markov Models. *Statistics*, 26(4):345–354, 1995.
19. P. R. Tadikamalla. Random Sampling from the Exponential Power Distribution. *Journal of the American Statistical Association*, 75:683–686, 1980.
20. M. E. Tipping and C. M. Bishop. Mixtures of Probabilistic Principal Component Analyzers. *Neural Computation*, 11(2):443–482, 1999.